

# Revista Eletrônica de Sistemas de Informação

## ISSN 1677-3071

**v. 16, n. 3**

set-dez 2017

DOI: <https://doi.org/10.21529/RESI.2017.1603>

### Sumário

#### Software livre

##### O MODELO ABERTO DE DESENVOLVIMENTO DE SOFTWARE EM PEQUENAS EMPRESAS BRASILEIRAS

Luiz Fernando Albertin Bono Milan, Fernando de Souza Meirelles

 [10.21529/RESI.2017.1603001](https://doi.org/10.21529/RESI.2017.1603001)

#### Governança de TI

##### GOVERNANÇA DE TECNOLOGIA DA INFORMAÇÃO: A MEIO CAMINHO ENTRE O ISOMORFISMO E A COMODITIZAÇÃO

Pedro Jácome De Moura Jr

 [10.21529/RESI.2017.1603002](https://doi.org/10.21529/RESI.2017.1603002)

#### Fast Track Semead

##### APLICAÇÃO DO MODELO UTAUT NA UNIVERSIDADE FEDERAL DE RONDÔNIA: UM ESTUDO SOBRE A ACEITAÇÃO E UTILIZAÇÃO DE SISTEMA DE INFORMAÇÃO DE GESTÃO ACADÊMICA

Anderson Pinheiro da Silva, Carolina Yukari Veludo Watanabe

 [10.21529/RESI.2017.1603003](https://doi.org/10.21529/RESI.2017.1603003)



Este trabalho está licenciado sob uma [Licença Creative Commons Attribution 3.0](https://creativecommons.org/licenses/by/3.0/).

Esta revista é (e sempre foi) eletrônica para ajudar a proteger o meio ambiente, mas, caso deseje imprimir esse artigo, saiba que ele foi editorado com uma fonte mais ecológica, a *Eco Sans*, que gasta menos tinta.

*This journal is (and has always been) electronic in order to be more environmentally friendly. Now, it is desktop edited in a single column to be easier to read on the screen. However, if you wish to print this paper, be aware that it uses Eco Sans, a printing font that reduces the amount of required ink.*

# O MODELO ABERTO DE DESENVOLVIMENTO DE SOFTWARE EM PEQUENAS EMPRESAS BRASILEIRAS

## THE OPEN SOURCE MODEL OF SOFTWARE DEVELOPMENT IN SMALL BRAZILIAN COMPANIES

(artigo submetido em agosto de 2017)

**Luiz Fernando Albertin Bono Milan**

Doutorando da Escola de Administração de Empresas da Fundação Getulio Vargas

(EAESP-FGV)

luiz.milan@gvmail.br

**Fernando de Souza Meirelles**

Professor da Escola de Administração de Empresas da Fundação Getulio Vargas

(EAESP-FGV)

fernando.meirelles@fgv.br

### ABSTRACT

*Large companies in the information technology sector are opening the codes of strategic products to gain a competitive advantage. Under the philosophical paradigm of Critical Realism, this study aims to identify the mechanisms involved in the decision to adopt the open model of software development by small Brazilian companies. Using multiple case studies, the data were obtained from four Brazilian technology companies. The qualitative material collected in 2017 was analyzed using inductive and deductive methods, seeking evidence to corroborate the existing literature and identifying mechanisms associated with the specific context explored in this study. The main contributions of the study are the level of analysis, its anchoring in empirical data, the chosen context and the philosophical paradigm used by the authors. The study found that political instability and the high cost associated with adopting the open model of software development could be barriers for small companies wishing to adopt this model. The study also allows market managers to reassess their software development strategies.*

*Key-words: open source software; open source as business strategy; critical realism; multiple case study.*

### RESUMO

Grandes empresas do setor de tecnologia da informação estão abrindo os códigos de produtos que são estratégicos para se obter vantagem competitiva. Sob o paradigma filosófico do Realismo Crítico, este estudo tem o objetivo de identificar os mecanismos envolvidos na decisão de adoção do modelo aberto de desenvolvimento de programas de computador por pequenas empresas brasileiras. Utilizando estudo de casos múltiplos, os dados contemplam quatro empresas brasileiras de tecnologia. O material qualitativo coletado em 2017 foi analisado utilizando métodos indutivos e dedutivos, buscando evidências para corroborar a literatura existente e identificar eventuais mecanismos associados ao contexto específico explorado pelo estudo. As principais contribuições do estudo são o nível de análise adotado, sua ancoragem em dados empíricos, o contexto escolhido e o paradigma filosófico utilizado pelos autores. O estudo identificou que a instabilidade política e o elevado custo associado à adoção do modelo aberto de desenvolvimento de programas de computador podem ser barreiras para as pequenas empresas adotarem tal modelo. Aos gestores do mercado, permite a reavaliação de suas estratégias em relação ao desenvolvimento de seus programas.

Palavras-chave: programa de computador de código aberto; código aberto como estratégia empresarial; realismo crítico; estudo de caso múltiplo.

# 1 INTRODUÇÃO

Atualmente, a economia é altamente dependente de programas de computador. Tem sido cada vez mais comum empresas, que não comercializam programas, os desenvolverem para suportar seus negócios. Há ainda, previsões indicando que a necessidade de saber programar deve aumentar para carreiras que tradicionalmente não exigem tal habilidade do profissional.

Diante deste cenário, o aprofundamento dos estudos sobre a adoção do modelo aberto de desenvolvimento de *software* pelas empresas se mostra valioso. Considerável atenção foi dedicada a outros aspectos desse modelo no nível de análise do indivíduo, deixando sua adoção pelas empresas carente de maior aprofundamento.

Os projetos de programas de código aberto migraram de comunidades de desenvolvedores individuais para comunidades de organizações comerciais, formadas principalmente por pequenas e médias empresas (ÅGERFALK E FITZGERALD, 2008).

Grandes empresas do setor de tecnologia da informação estão abrindo os códigos de produtos que são potencialmente estratégicos. Entre elas estão Amazon ([github.com/gluon-api/gluon-api](https://github.com/gluon-api/gluon-api)), Google ([www.tensorflow.org](http://www.tensorflow.org)), Facebook ([pytorch.org](http://pytorch.org)) e Microsoft ([www.onnx.ai](http://www.onnx.ai)), que abriram os códigos de suas soluções de inteligência artificial. Abrir o código desse tipo de *software*, que se revela capaz de oferecer vantagem competitiva a uma empresa, pode sugerir retorno financeiro nulo, porém o que provavelmente essas empresas estão esperando, como retorno, é a oportunidade de ser a base de futuras inovações (SHAFTO, 2016).

Devido a essa nova abordagem estratégica das empresas de tecnologia em relação ao *software* de código aberto, a investigação dos mecanismos que levam as empresas a abrirem seus códigos torna-se relevante. Essa abordagem pode ser vantajosa não só para a indústria de tecnologia, mas, também, para outros setores. Porém, a adoção do modelo aberto de desenvolvimento de *software* por pequenas empresas brasileiras pode ser desafiador.

Há avanços nos modelos que explicam os fatores que levam os desenvolvedores e os usuários a se envolverem com programas de código aberto (FANG E NEUFELD, 2009; KE E ZHANG, 2010; VON KROGH *et al.*, 2012; HOWISON E CROWSTON, 2014). Também há estudos que suportam o modelo que endereça os fatores determinantes da atratividade dos projetos de código aberto (SANTOS *et al.*, 2013). Porém, não foram identificados, na literatura sobre o tema, os mecanismos envolvidos na decisão dos gestores de pequenas e médias empresas brasileiras para adotar o modelo aberto de desenvolvimento de *software*, seja ao abrir o código-fonte de seu produto ou serviço de *software* ou alocando seus recursos no desenvolvimento de *software* de código aberto, seja desenvolvendo software sobre o qual a empresa não tem propriedade intelectual.

Em grande parte, os estudos sobre o modelo aberto de desenvolvimento de *software* focam o nível do indivíduo, assim como os estudos de Steinmacher *et al.* (2014) e Rousinopoulos, Robles e González-Barahona (2014).

Este estudo tem o objetivo de identificar mecanismos envolvidos na adoção do modelo aberto de desenvolvimento de *software* por pequenas empresas brasileiras, os desafios e as experiências de empresas que se envolveram com este modelo de desenvolvimento.

## 2 REVISÃO DA LITERATURA

Para o levantamento dos artigos publicados, reconhecidos e referenciados por esta audiência, foram inicialmente identificados os artigos que endereçam o tema software de código aberto até o ano de 2016, em todos os periódicos do *senior scholars' basket of journals* (AIS, 2011), importante conjunto de periódicos científicos na área de gestão de Sistemas de Informação. Os termos utilizados na busca foram: *open source software*; OSS; FLOSS; *free software*; e, *open source development*. A seguir, foram selecionados os artigos mencionados por estes e, a cada novo artigo, suas citações para outros que endereçam os temas mapeados, assim procedendo até não haver mais artigos sem identificação. Foram identificados 438 artigos em 144 periódicos científicos. Apenas artigos de periódicos científicos foram selecionados. Não foram considerados artigos de congressos ou outros tipos de publicação científica ou comercial.

A literatura de *software* de código aberto já abordou os projetos que utilizam esse tipo de desenvolvimento de programas sob diferentes perspectivas. Artigos já endereçaram o sucesso de projetos de código aberto sob diferentes pontos de vista (HERBSLEB E MOCKUS, 2003; GREWAL *et al.*, 2006; LONG, 2006; STEWART E GOSAIN, 2006; COMINO *et al.*, 2007; SUBRAMANIAM *et al.*, 2009), questões referentes aos tipos de licenças utilizadas para distribuição dos programas desenvolvidos (CARVER, 2005; LERNER E TIROLE, 2005; STEWART *et al.*, 2006; KUEHNEL, 2008; TSAI, 2008), comparação entre programas de código aberto e fechado (GAMBARDELLA E HALL, 2006; SEN, 2007; LIN, 2008; CHENG *et al.*, 2011), fatores que levam desenvolvedores e usuários a se envolverem com esses projetos (FANG E NEUFELD, 2009; KE E ZHANG, 2010; VON KROGH *et al.*, 2012; HOWISON E CROWSTON, 2014), fatores que determinam a atratividade de um projeto de código aberto (SANTOS *et al.*, 2013), fatores que determinam a adoção desses software (CHAU E TAM, 1997; KAUFFMAN E MOHTADI, 2004; GWEBU E WANG, 2011), governança em projetos de código aberto (O'MAHONY E FERRARO, 2007; MIDHA E BHATTACHERJEE, 2012; DI TULLIO E STAPLES, 2013), entre outras abordagens.

Pesquisadores, frequentemente, apontam para um grupo de fatores que podem influenciar a decisão de uma empresa de abrir o código de seu *software* para o público. Os fatores encontrados na literatura apontam que



as empresas podem ser levadas a abrir o código para melhorar a qualidade de seu *software*, elevar a possibilidade de inovação, diminuir a vantagem do concorrente, diminuir os custos com desenvolvimento de *software*, risco do concorrente se apropriar do código, idade e tamanho da empresa e tipo de aplicação. Nas seções seguintes é aprofundada a revisão referente a cada um destes fatores separados em três categorias, que são: questões técnicas, questões relacionadas a custos e questões relacionadas à vantagem competitiva.

Apesar de os projetos de programas de código aberto normalmente não seguirem os mesmos processos de gestão de qualidade dos projetos de programas de código fechado, alguns fatores, como o tamanho da comunidade de usuários e desenvolvedores envolvidos, podem levar a produtos de alta qualidade (STAMELOS *et al.*, 2002; ABERDOUR, 2007; SANTOS *et al.*, 2013). Os padrões de excelência que emergem de comunidades globais de desenvolvedores de *software* de código aberto podem ganhar amplo apoio e impactar os padrões de qualidade que os usuários e clientes esperam. Exemplos destes padrões de excelência são a revisão do código realizada por pares e os processos rápidos de resposta no desenvolvimento de *software* de código aberto. Isso leva à cultura de “fazer a coisa certa” em vez de “criar remendos rapidamente” e pode resultar em códigos de melhor qualidade (VON KROGH *et al.*, 2012). Essas questões podem levar empresas a considerar a possibilidade de abrir o código de seus programas visando a um ganho na qualidade de seu produto e/ou serviço.

Foi utilizada como arcabouço teórico para abordar a decisão de uma empresa em adotar o modelo aberto de desenvolvimento de *software*, a literatura sobre inovação aberta (*open innovation* em inglês). Essa escolha se deu devido à sua aderência ao fenômeno que ainda não conta com literatura específica que explique sua adoção pelas empresas.

O conceito de inovação aberta foi inicialmente apresentado por Chesbrough (2003), que destacou fatores que corroeram os limites de onde a inovação ocorre e catalisaram um movimento em direção a modelos de inovação mais abertos.

A definição de inovação aberta (IA) se transformou desde sua apresentação inicial. Passou a ser considerada um processo de inovação distribuído, baseado em fluxos de conhecimentos gerenciados propositalmente pelas fronteiras organizacionais, em que a IA é essencialmente um conceito que reside no nível da organização. Ainda, o alinhamento do processo de IA com o modelo de negócios da organização passou a ser considerado. Enquanto o conceito original de IA é centrado na empresa, a literatura o vincula a vários fenômenos de inovação relacionados, como usuários como inovadores, comunidades de inovação ou desenvolvimento de *software* de código aberto, que não consideram necessariamente a empresa como o nível focal de análise (BOGERS *et al.*, 2017).

West & Gallagher (2006) apresentam estratégias para o desenvolvimento de *software* de código aberto como solução para os desafios da

IA. Defendem ainda que a literatura de IA ajuda a explicar como as empresas usam o aumento do modelo aberto de desenvolvimento de *software* para criar novas estratégias de inovação. O uso do desenvolvimento de código aberto pelas empresas normalmente começa de maneira que não altera seu modelo de negócios fundamental ou ocorre quando um modelo de negócios existente é tão ameaçado que a empresa é forçada a fazer mudanças drásticas.

O modelo aberto de desenvolvimento de *software* é considerado uma forma de IA (GASSMANN *et al.*, 2010), dessa forma, para abordar sua adoção pelas empresas, além dos construtos identificados na literatura sobre desenvolvimento de *software* de código aberto, foi utilizada a literatura de adoção de inovação aberta.

Um vasto conjunto de estudos empíricos identificou fatores com coeficientes estatísticos significantes correlacionados com a adoção da IA (SCHROLL E MILD, 2012), são eles: intensidade de P&D, tamanho da empresa, amplitude estratégica e diversificação, agressividade tecnológica, capacidades de projeto, orientação para o cliente e aumento dos custos de desenvolvimento.

### 3 MÉTODO DE PESQUISA

O presente estudo adota o paradigma filosófico denominado Realismo Crítico (RC). O RC está emergindo rapidamente como uma alternativa paradigmática viável para a realização de pesquisas em ciências sociais, adequado para o desenvolvimento de explicações causais de fenômenos complexos (WYNN JR E WILLIAMS, 2012). Ele fornece uma terceira forma de pensamento entre os polos do positivismo e do interpretativismo, pois utiliza elementos e pontos fortes de ambos, permitindo que os pesquisadores gerem teorias que forneçam explicações causais detalhadas de fenômenos complexos em sistemas abertos (MARKUS E SILVER, 2008; REED, 2009; WYNN JR E WILLIAMS, 2012).

As origens do RC estão associadas ao trabalho seminal de Bhaskar (1975) e seu desenvolvimento recebeu atenção de outros pesquisadores (COLLIER, 1994; LÓPEZ E POTTER, 2005; PEARCE E FRAULEY, 2007; SAYER, 2010). Sua utilização como postura filosófica para pesquisa no campo de administração de TI tem ganhado relevância (MUTCH, 1999; MINGERS, 2001; MORTON, 2006; DOBSON *et al.*, 2007; KHOO E ROBEY, 2007).

Sendo uma forma particular de realismo, o RC reconhece o mundo e as entidades que constituem a realidade, independentemente do conhecimento humano ou a capacidade de percebê-los. Essa independência não depende de qualquer conhecimento direto ou crenças subjetivas quanto à existência de qualquer entidade. Em vez disso, o RC reconhece que o mundo não é facilmente redutível às nossas percepções e experiências. Em outras palavras, a natureza da realidade não é facilmente compre-

endida, caracterizada ou medida, o que significa que os seres humanos experimentam apenas uma parte dela (WYNN JR E WILLIAMS, 2012).

Zachariadis, Scott, & Barrett (2012) explicam que o RC confronta simultaneamente as preocupações centrais dos regimes das ciências naturais e sociais. Isso faz com que o RC seja particularmente interessante para o estudo de sistemas de informação, que tem relevância significativa para a ciência natural (devido às suas características tecnológicas) e ciências sociais (devido às suas aplicações em contextos profundamente humanos, como as organizações).

O objetivo de um estudo baseado no RC é explicar os mecanismos que geram um determinado evento, mais do que a capacidade de fazer previsões sobre eventos futuros ou de compreender os significados socioculturais por trás dos eventos. Uma explicação estipula os fatores que se presume causarem um determinado resultado (YIN, 2014) e identifica os mecanismos que emergem dos componentes de uma estrutura física e social que produzem o evento de interesse (SAYER, 2010). Ainda, segundo Bhaskar (2013), a crença na existência de um mecanismo pode ser baseada na nossa capacidade de observá-lo diretamente (critérios perceptivos), com ou sem ferramentas para isso, ou na capacidade de observar seus efeitos (critérios causais).

Os princípios metodológicos do RC se baseiam na explicação de eventos, estruturas e contexto, abdução (identificação e elaboração em relação aos poderes/tendências da estrutura que possam ter interagido para gerar os eventos explicados), corroboração empírica e emprego de triangulação e multimétodos (WYNN JR E WILLIAMS, 2012).

### 3.1 ESTUDO DE CASOS MÚLTIPLOS

Apesar de não haver uma metodologia de pesquisa específica associada ao paradigma RC, pesquisadores identificaram o método de estudo de caso como a melhor abordagem para explorar a interação entre estrutura, eventos, ações e contexto, para identificar e explicar mecanismos causais. Um estudo de caso investiga um ou um pequeno número de entidades sociais ou situações em que os dados são coletados usando múltiplas fontes. Um estudo de caso é uma investigação empírica, que estuda um fenômeno contemporâneo em profundidade e dentro do contexto da vida real, especialmente quando as fronteiras entre fenômeno e contexto não são claramente evidentes (MILES E HUBERMAN, 1994; ACKROYD, 2010; EASTON, 2010; WYNN JR E WILLIAMS, 2012; YIN, 2014).

O presente estudo utiliza como principal método para coleta de dados, o estudo de casos múltiplos. A escolha dos casos foi realizada buscando oferecer uma amplitude satisfatória de representação do contexto estudado. Dessa forma, foram escolhidas, como casos de estudos, empresas pequenas, que, em relação à adoção do modelo aberto de desenvolvimento de *software*, tiveram sucesso, fracassaram, começaram com o modelo aberto e migraram para o modelo fechado ou consideraram adotar o modelo aberto, mas não o fizeram.



### 3.2 IDENTIFICAÇÃO DOS CASOS E COLETA DO MATERIAL QUALITATIVO

Para identificar possíveis casos, foram utilizadas algumas técnicas. Inicialmente foram procuradas notícias na mídia brasileira e em grupos de interesse em redes sociais, como LinkedIn e Facebook, referindo-se às empresas que haviam aberto o código de um ou mais de seus programas. Após a identificação das empresas, foram utilizadas plataformas de hospedagem de repositórios de código-fonte (ex.: Github) para identificar os repositórios dessas empresas. Esses repositórios são utilizados pelas empresas que abrem os códigos de seus programas para tornar o código acessível a qualquer pessoa. Então, após a localização dos repositórios das empresas, foi possível identificar profissionais que trabalhavam nas respectivas empresas e estavam envolvidos com o desenvolvimento desses programas. Com a identificação dos profissionais foi possível localizar seus contatos em redes sociais como LinkedIn, About.me, *websites* pessoais dos profissionais e seus perfis no Facebook. Finalmente, com os contatos dos profissionais, a abordagem a eles foi iniciada, buscando identificar quais deles estariam dispostos a falar sobre a decisão de abertura do código do *software* identificada previamente, caso eles tivessem feito parte da decisão. Caso contrário, era solicitado o contato de profissionais que participaram da decisão. O processo de identificação das empresas encerrou-se ao serem identificadas empresas que forneciam amplitude satisfatória de representação do contexto estudado.

Após a identificação da empresa, foram iniciadas as entrevistas. As entrevistas ocorreram durante o segundo semestre de 2017 e tiveram duração entre 30 e 40 minutos. Foram realizadas por meio de serviços de videoconferência, como Google Hangouts e Skype, gravadas e transcritas com auxílio da *Application Programming Interface* (API) de *Automatic Speech Recognition* (ASR) da plataforma *Google Cloud* para posterior análise.

Além das entrevistas, também foram coletadas notas realizadas durante as entrevistas, mensagens de e-mail trocadas com questionamentos para os entrevistados e outros profissionais que não foram entrevistados e gravação de sessão de negociação a respeito da eventual abertura do código-fonte do principal *software* de uma pequena empresa que desenvolve sistema de gestão.

### 3.3 DESCRIÇÃO DOS CASOS

Como indicado por Wynn Jr & Williams (2012), os casos selecionados são descritos nas seções seguintes, incluindo ações e resultados, além da abstração da sequência de eventos, buscando incluir as experiências dos participantes e observadores. Todos os casos selecionados para o estudo estão localizados no Brasil, apesar de um ter atuação internacional.

#### 3.3.1 Caso Empresa de Ferramenta para Colaboração

A empresa fundada em 2015 oferece ao mercado uma solução para a comunicação entre membros de equipes. Sua solução é uma plataforma

de bate-papo baseada na web com funcionalidades inovadoras e tem o código aberto desde o início de seu desenvolvimento. Em 2016, um dos maiores fundos de investimentos especializados em negócios na área de Tecnologia da Informação (TI), o *New Enterprise Associates* (NEA), investiu cinco milhões de dólares norte-americanos na empresa. No segundo semestre de 2017 a empresa tinha 23 funcionários, a maioria trabalhando remotamente.

O fundador da empresa teve uma experiência positiva quando criou um projeto de *software* de código aberto em 2000. O projeto criado chamou a atenção de profissionais nos EUA. Um desses profissionais veio ao Brasil e o contratou para desenvolver funcionalidades específicas para o *software*. Essa experiência fez com que ele, o fundador da empresa, quisesse repetir a criação de *software* de código aberto, porém desenvolveu seu negócio com *software* de código fechado, que oferece ao mercado uma solução de *Customer Relationship Management* (CRM).

Devido à demanda de seus clientes por uma solução de bate-papo em tempo real integrado a seu CRM e diante de restrições de recursos para seu desenvolvimento, ele acreditou que desenvolver o *software* de bate-papo no modelo aberto poderia atrair pessoas para testá-lo e, ainda, contribuir com novas funcionalidades.

Como sócio majoritário, o fundador da empresa decidiu de forma unilateral, criar o *software* de bate-papo com o código aberto sob a licença MIT. Após sua decisão, a empresa demorou seis meses para desenvolver um protótipo e divulgar seu código sob uma licença de software aberto. Apesar de haver dezenas de empresas oferecendo *software* de bate-papo com serviço na Internet, o deles era um dos primeiros de código aberto. Isso atraiu a atenção de empresas e fundos de investimentos um mês após a publicação do código. Após um ano, recebendo propostas de empresas e fundos de investimentos para aquisição de parte ou do controle total de sua empresa, em 2016, os sócios aceitaram criar uma nova empresa apenas para dar segmento no desenvolvimento do *software* aberto de bate-papo, permitindo, assim, o aporte de um dos maiores fundos de investimentos em empresas de TI, nessa nova empresa.

### 3.3.2 Caso Empresa de Sistema de Gestão

A empresa desenvolve *software* de gestão para o segmento de recuperação de crédito. Foi fundada em 2014 e teve um faturamento de aproximadamente trezentos mil reais, em 2017. A empresa possui seis funcionários. Entre seus clientes, estão empresas de recuperação de crédito e departamentos de grandes empresas brasileiras e multinacionais.

Antes de fundarem a empresa, os sócios tinham outra empresa que, durante dezesseis anos, operou utilizando quase somente software de código aberto. Entre eles estavam sistemas operacionais dos servidores e estações de trabalho, editores de texto, planilhas eletrônicas, sistemas gerenciadores de bancos de dados, linguagens de programação, sistema

de telefonia e soluções de infraestrutura de redes de computadores. Apesar do amplo uso de programas de código aberto na antiga empresa, os sócios nunca tiveram a experiência de abrir o código de um *software* da própria empresa.

Após fundarem a atual empresa, explorando um novo modelo de negócio para o ramo de recuperação de crédito sem obterem retorno financeiro satisfatório, os sócios decidiram explorar comercialmente o *software* desenvolvido por eles na antiga empresa. Então passaram a oferecer o *software* como serviço na Internet, baseado em computação em nuvem.

Um dos sócios acredita que a abertura do código do *software* oferecido pela empresa pode trazer benefícios, porém os outros sócios não acreditam que a empresa opere em um mercado em que o modelo aberto de desenvolvimento possa trazer vantagens. Eles acreditam que o mercado de recuperação de crédito poderia utilizar o *software* sem oferecer retorno algum para a empresa.

Além de não acreditarem em retorno pela adoção do modelo aberto de desenvolvimento de seus programas, os sócios expressam preocupação com o custo de preparação de todo código para poder abri-lo. Até o segundo semestre de 2017, a empresa não havia aberto algum código de seus programas.

### 3.3.3 Caso Empresa de Projetos de *Software*

A empresa de projetos de *software* foi fundada em 1987 por um cientista da computação, que começou sua carreira em um centro de pesquisas em telecomunicações de uma empresa estatal, desenvolvendo compiladores e protocolos de comunicação. No início de suas atividades (o caso selecionado), desenvolvia protocolos de comunicação para automação bancária. Com o passar do tempo, a empresa deixou de desenvolver programas básicos, como pequenos sistemas operacionais para desenvolver aplicações que davam suporte às atividades bancárias. A experiência com o desenvolvimento de aplicações levou a empresa a se especializar no desenvolvimento de programas customizados.

Em 2013, a empresa de projetos de *software* foi comprada pelo valor de nove milhões de reais por uma empresa que oferece sistemas de *Enterprise Resource Planning* (ERP) para o mercado varejista. A empresa compradora faturou, em 2017, aproximadamente seiscentos milhões de reais e tinha próximo de três mil funcionários.

O fundador do caso, no ano 2000, com o objetivo de consolidar um *software* engenho de *workflow* no mercado brasileiro, para geração de nova fonte de receita, resolveu adotar o modelo aberto de desenvolvimento. O desenvolvimento do *software* começou em 2002 sob a licença GNU General Public License (GPL), ano em que, em função das incertezas políticas devido à aproximação das eleições presidenciais, o país passou por uma crise, que perdurou até 2003. Como boa parte da carteira de

clientes da empresa era do setor financeiro, houve significativa redução do faturamento e consequente cancelamento de projetos desse setor. Dessa forma, o capital utilizado para as iniciativas da empresa, como o *software* engenho de *workflow*, foi restringido. Com o esgotamento dos recursos financeiros, a iniciativa de código aberto da empresa foi abandonada. O código desse *software* ainda estava disponível no repositório de projetos de código aberto SourceForge.net, em 2017, porém não conseguiu atrair a colaboração de desenvolvedores externos à empresa. Após essa experiência sem sucesso com o modelo aberto de desenvolvimento de *software*, até o segundo semestre de 2017, a empresa não abriu o código de algum outro *software*.

Um dos profissionais que trabalhava na equipe de desenvolvimento do *software* engenho de *workflow*, com o abandono do projeto, solicitou à empresa autorização para utilizar seu código para abrir sua própria empresa e explorá-lo no modelo fechado de desenvolvimento. Sua iniciativa foi bem-sucedida e, em 2017, o *software* ainda era um dos produtos oferecidos ao mercado pela nova empresa.

#### 3.3.4 Caso Empresa de Ferramenta para Gestão de Projetos e Processo

A empresa iniciou suas operações em 2001 com três pessoas. Em 2017, ela possuía nove funcionários e faturava entre três e cinco milhões por ano. Seus produtos são programas utilizados como ferramenta para auxiliar na gestão de projetos e processos. Entre seus clientes estão empresas, institutos e fundações como o CPqD, Instituto Genius de Tecnologia, Embraer, Samsung e bancos nacionais.

Seu início foi explorando um *software* de código aberto que dois dos fundadores da empresa haviam ajudado a desenvolver, o que levou a empresa a ser contratada por um centro de pesquisa e desenvolvimento de tecnologias para desenvolver novas funcionalidades para o próprio *software*. A empresa conseguiu, ainda, o financiamento da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) para continuar o desenvolvimento do *software*.

Após esse início, a empresa passou a buscar clientes que não eram desenvolvedores de tecnologias ou agências de fomento. Nesse momento, em meados da primeira década do século atual, a empresa identificou que seu *software* não era facilmente aceito devido à licença GNU *General Public License* (GPL), que obriga outros programas que utilizam seu código a serem licenciados sob os mesmos termos. Dessa forma a empresa decidiu desenvolver uma versão do *software* com o código fechado para atender a esses clientes.

Após ter adotado o modelo fechado de desenvolvimento de *software*, até o segundo semestre de 2017 a empresa não mais havia desenvolvido iniciativas no modelo aberto, apesar de já ter cogitado abrir o código de algumas de suas ferramentas.



### 3.4 ANÁLISE DO MATERIAL QUALITATIVO COLETADO

Sob o paradigma filosófico do RC, para conseguir a explicação mais próxima do que acontece nessa realidade independente, é preciso utilizar formas de triangulação, entre elas a triangulação dos métodos utilizados para efetuar a análise dos dados coletados (WYNN JR E WILLIAMS, 2012). Desse modo, o presente estudo utiliza dois conjuntos de técnicas para a análise qualitativa do material coletado: a análise dirigida de conteúdo (*directed content analysis*) e a análise fundamentada (*grounded analysis*).

Os dois conjuntos de técnicas foram escolhidos por fornecerem características fundamentais distintas. A análise fundamentada tem uma característica indutiva, ou seja, é buscado nos dados códigos e categorias sem considerar qualquer teoria pré-existente (STRAUSS E CORBIN, 1998), enquanto a análise dirigida de conteúdo tem uma característica dedutiva, ou seja, os códigos e categorias partem da teoria e seus resultados oferecem evidências de apoio e/ou não sustentação para uma teoria (HSIEH E SHANNON, 2005).

A análise fundamentada seguiu os procedimentos de codificação, aberta, axial e seletiva descritos por Strauss & Corbin (1998, cap. 8, 9 e 10). Strauss & Corbin (1998, cap. 8) definem a codificação aberta como o processo analítico por meio do qual os blocos de construção da teoria, suas características e intervalo de variação são identificados. A codificação axial é definida por Strauss & Corbin (1998, cap. 9) como o processo de relacionar categorias às suas subcategorias, é chamado de "axial", pois a codificação ocorre em torno do eixo de uma categoria, vinculando categorias ao nível de propriedades e dimensões, ajudando assim, a integrar o contexto condicional em que se situa um bloco de construção da teoria (categoria) com a sequência de ação e/ou interação pertencentes a um fenômeno à medida que evoluem ao longo do tempo. Na codificação aberta, a preocupação é com a geração de categorias e suas propriedades e, em seguida, busca-se determinar como as categorias variam em sua dimensão. Na codificação axial, as categorias são sistematicamente desenvolvidas e vinculadas às subcategorias. A codificação seletiva é o processo de integração e refinamento das categorias (STRAUSS E CORBIN, 1998, cap. 10).

A codificação aberta foi iniciada com a rotulação, onde cada trecho (frases e/ou parágrafos) das transcrições, textos coletados na mídia e e-mails trocados com entrevistados recebeu um rótulo. Este processo foi iniciado após as primeiras entrevistas serem transcritas. Junto com a rotulação dos trechos analisados, foram feitas anotações (*memo*) referentes aos rótulos que ajudavam a explorar novos pontos nas entrevistas futuras.

Utilizando os rótulos atribuídos aos trechos do material analisado, foram criadas categorias que endereçam problemas, preocupações e questões que são importantes para o fenômeno identificado. Uma vez que uma categoria é identificada, torna-se mais fácil lembrá-la, realizar ponderações e desenvolvê-la quanto às suas propriedades e dimensões e, assim,



diferenciá-la e dividi-la em subcategorias, que, por sua vez, especificam uma categoria indicando informações de como, quando, onde e porque um fenômeno provavelmente ocorre. Subcategorias, como as categorias, também possuem suas propriedades e dimensões.

Na codificação axial, foram relacionadas as categorias às suas subcategorias em relação às suas propriedades e dimensões, com objetivo de identificar como, quando, onde, porque e quais as consequências das relações entre estruturas e processos. Nesta etapa foram identificadas condições macro e micro, ou seja, macro, mais distantes da fonte de ação e/ou interação e, micro, mais próximas da fonte de ação e/ou interação. Assim, buscou-se a relação entre as ações e/ou interações com suas consequências. As ações e interações denotam o que está ocorrendo ou deixando de ocorrer com os indivíduos, grupos e organizações, como isso está acontecendo e em qual contexto.

As categorias principais foram identificadas na codificação seletiva e uma principal foi definida. Ainda na codificação seletiva, foram utilizados mapas mentais para testar a consistência das principais categorias e seus relacionamentos em esquemas amplos e distintos, buscando qual esquema formava a melhor explicação/teoria para os dados coletados. Para garantir a validade do esquema teórico desenvolvido, ele foi confrontado com os dados brutos, para verificar se era capaz de explicar a maioria dos casos.

Como indicado por Elo & Kyngäs (2008), a análise dirigida de conteúdo, abordagem dedutiva, foi baseada nos conceitos identificados na revisão da literatura do estudo, buscando ir do geral para o específico. As fases da análise seguiram a orientação de Elo & Kyngäs (2008).

Ambas as análises foram realizadas com o auxílio do software RQDA (*R package for Qualitative Data Analysis*).

## 4 ANÁLISE

A abordagem utilizada no estudo permitiu a identificação de mecanismos causais, envolvendo aspectos técnicos, estruturais e sociais presentes no fenômeno pesquisado. Foi possível identificar nos casos, fatores endereçados pela literatura e outros potencialmente envolvidos na adoção do modelo aberto de desenvolvimento de *software* pelas pequenas e médias empresas no Brasil.

Abrir o código de um *software* e receber contribuições sem a necessidade de remunerá-las pode levar ao pensamento de imediata redução de custo de desenvolvimento, porém parece haver custo para a abertura do código. Este custo inicial, gerado devido à necessidade de adaptação do código para sua exposição ao público em geral e fomento de uma comunidade para aprimorá-lo, pode ser uma barreira para as pequenas empresas brasileiras.

*"Vocês têm, aí na Empresa alguma iniciativa de código aberto? Nossa, não. A gente é muito pequeno, a gente sempre foi muito pequeno. Então não temos condições para isso não."* (Fundador – Empresa de Ferramentas para Gestão de Projetos e Processo – Em transcrição de entrevista)

As empresas parecem enxergar no modelo aberto de desenvolvimento, uma forma de melhorar a qualidade do *software*, uma vez que mais pessoas podem examiná-lo e testá-lo. Todavia, acreditam que isso apenas ocorre quando conseguem criar uma comunidade engajada em seu desenvolvimento.

*"Sim, haverá melhora do produto, pois vamos melhorar uma parte fundamental do produto, seu código, só para abri-lo já é necessária uma revisão completa, ainda, com outras pessoas olhando o código, identificaremos falhas mais rapidamente. Se nos empenharmos em criar uma comunidade ativa ao redor dele, evidente."* (CTO – Empresa de Sistema de Gestão – Em transcrição de sessão de negociação entre sócios)

A experiência positiva do desenvolvedor que já participou de projetos de código aberto pode encorajar iniciativas similares. Ainda, a experiência adquirida pode trazer conhecimentos que favoreçam o melhor aproveitamento comercial do modelo aberto de desenvolvimento.

*"pois até então esta empresa não tinha lançado nada open source, eu tinha tido uma experiência muito boa anos atrás, em que criei um projeto open source, minha empresa na faculdade. Muitas pessoas começaram a baixar nos Estados Unidos e, meses depois, veio um americano que estava criando uma startup, especificamente para contratar a minha empresa."* (CEO – Empresa de Ferramenta de Colaboração – Em transcrição de entrevista)

A evidência apresentada foi coletada em entrevista com a pessoa que após sua primeira experiência bem-sucedida com o modelo aberto de desenvolvimento, conseguiu atrair investimento substancial para sua segunda iniciativa com este modelo de desenvolvimento.

Conseguir atrair novos talentos, dispostos a experimentar novos modelos de desenvolvimento e potencial disposição para inovação, parece levar as empresas a perceber algum valor na adoção do modelo aberto.

*"Participamos de uma feira na Unicamp e quando nosso futuro funcionário viu que o projeto era open source, ele foi lá e falou: - bom... eu não entendi bem, mas vocês vão me contratar, porque é com isso que eu quero trabalhar. Ou seja, ele não perguntou o que era o projeto, o que ele ia aprender, que linguagem de programação a gente ia usar, ele não perguntou nada disso. Ele falou: -open source, eu quero."* (Fundador - Empresa de Projetos de Software – Em transcrição de entrevista)

Abrir o código de um *software* que foi desenvolvido no modelo fechado por algum tempo, pode gerar maior resistência do que iniciar o desenvolvimento já com o modelo aberto, pois adotando o modelo aberto

no início evita o custo necessário à adaptação do código, apesar de não eliminar o custo envolvido no fomento da comunidade de desenvolvedores.

*“Supondo que a gente consiga criar uma comunidade para ajudar no desenvolvimento do software, quanto tempo levaríamos para recuperar o gasto necessário para abriremos o código do sistema? Se fosse aberto desde o início, não teríamos o custo que teríamos hoje para abrir o código dele.”* (COO – Empresa de Sistema de Gestão – Em transcrição de sessão de negociação entre sócios)

Há evidência de que, além de suportar o receio de o concorrente utilizar o código-fonte, apontam para o receio do uso oportunista do código, ou seja, sem nenhuma contrapartida.

*“Tá, mas se abriremos o código de nosso sistema, o que vamos fazer depois? Como vamos gerar receita se todos vão usar sem pagar? E novos concorrentes que só vão precisar acionar suas áreas de venda para vender algo que já está pronto para eles comercializarem? Ainda, quem vai pagar por um software que é distribuído gratuitamente? Duvido até mesmo que alguém contribuiria com o código”* (CEO – Empresa de Sistema de Gestão – Em transcrição de sessão de negociação entre sócios)

A contrapartida em projetos desenvolvidos sob o modelo aberto de desenvolvimento parece ser algo esperado. A pessoa ou empresa que utiliza um determinado *software* de código aberto parece sentir a necessidade de oferecer uma contrapartida a este uso, comumente em forma de colaboração com a melhoria do *software*, ou seja, serviço de desenvolvimento.

*“Mesmo sendo parceiros da Microsoft, utilizávamos muito software open source em nosso provedor de Internet naquela época. Estes softwares nos encantavam e nos davam vontade de retribuir de alguma forma, então tentamos nossa primeira e única iniciativa com o modelo aberto de desenvolvimento.”* (Fundador - Empresa de Projetos de Software – Em transcrição de entrevista)

O desejo de retribuição, normalmente em forma de serviço de desenvolvimento ou oferta de um *software* aberto, apareceu nos quatro casos.

As evidências sustentam fatores identificados na revisão da literatura, tais como: influência do tamanho e/ou idade da empresa, desejo de elevar a qualidade do código, desejo da empresa de ser mais inovadora, necessidade de reduzir custos com desenvolvimento, receio de o concorrente usar o código. Entretanto, novos potenciais fatores foram identificados, são eles: receio de uso oportunista do código, custo para abrir o código e instabilidade política.

## 5 CONCLUSÃO

O modelo aberto de desenvolvimento de *software* tem mudado muito a forma como empresas de Tecnologia da Informação têm desenvolvido seus produtos e serviços. Cada vez mais, há empresas globais, como Apple

([developer.apple.com/opensource](https://developer.apple.com/opensource)), Microsoft ([github.com/Microsoft](https://github.com/Microsoft)), Google ([opensource.google.com](https://opensource.google.com)), IBM ([developer.ibm.com/code/open](https://developer.ibm.com/code/open)), Netflix ([netflix.github.io](https://netflix.github.io)) e muitas outras de TI, com iniciativas de código aberto. Mesmo empresas que normalmente não são reconhecidas como de TI, como Walmart ([github.com/walmartlabs](https://github.com/walmartlabs)) e BOSCH ([oss.bosch-cm.com](https://oss.bosch-cm.com)), por exemplo, estão começando a adotar o modelo aberto de desenvolvimento de *software*.

Esse fenômeno é abordado por este estudo sob a filosofia de pesquisa denominada Realismo Crítico (RC). O objetivo de um estudo baseado no RC é explicar os mecanismos que geram um determinado evento, mais do que a capacidade de fazer previsões sobre eventos futuros ou compreender os significados socioculturais por trás dos eventos. Uma explicação estipula os fatores que se presume causarem um determinado resultado (YIN, 2014), identifica os mecanismos que emergem dos componentes de uma estrutura física e social que produzem o evento de interesse (SAYER, 2010). Ainda, segundo Bhaskar (2013), a crença na existência de um mecanismo pode ser baseada na capacidade de observá-lo diretamente, com ou sem ferramentas para isso, ou em nossa capacidade de observar seus efeitos.

Para os pesquisadores que executam suas pesquisas sob o paradigma do RC, existe uma maior probabilidade de se produzir uma teoria que explique bem a realidade, se for utilizado o maior número possível de técnicas para investigar um determinado fenômeno (WYNN JR E WILLIAMS, 2012). Dessa forma, este estudo utiliza técnicas indutivas e dedutivas de análise de material qualitativo para endereçar o fenômeno estudado.

O estudo, com base na literatura de *software* de código aberto, na literatura de inovação aberta e na análise de material qualitativo provenientes de quatro estudos de caso, aprofunda o entendimento referente à adoção do modelo aberto de desenvolvimento de software por pequenas empresas brasileiras.

As principais contribuições do estudo são o nível de análise adotado para abordar o modelo aberto de desenvolvimento de *software*, sua ancoragem em dados empíricos e o paradigma filosófico utilizado pelos autores. Há poucos estudos que abordam o tema no nível organizacional e, quando o fazem, não oferecem evidências empíricas.

Foram identificadas evidências que suportam os fatores identificados pela revisão da literatura, tais como: influência do tamanho e/ou idade da empresa, desejo de elevar a qualidade do código, desejo da empresa de ser mais inovadora, necessidade de reduzir custos com desenvolvimento, e receio de o concorrente usar o código.

Dois fatores não identificados previamente na literatura utilizada como base para o estudo emergiram dos casos estudados, foram eles: o custo envolvido na adoção do modelo aberto de desenvolvimento pode ser uma barreira para as pequenas empresas, embora se mostre menor quando o desenvolvimento do *software* é iniciado no modelo aberto; e, a



instabilidade política, observada em muitos períodos da história recente do Brasil, foi o principal fator que levou um dos casos a abandonar o modelo aberto de desenvolvimento, pois seus clientes restringiram investimentos deixando a empresa descapitalizada e a obrigando a descontinuar algumas de suas iniciativas.

Não foram encontradas evidências que indiquem uma forma estruturada e formal de avaliação da adoção do modelo aberto de desenvolvimento, porém o material avaliado mostra que a criação de uma comunidade de desenvolvedores ao redor do *software* que teve seu código aberto, a atração da mídia e a oportunidade gerada de participação em fóruns profissionais de discussões, figuram entre os principais indicadores de avaliação.

## REFERÊNCIAS

ABERDOUR, M. Achieving quality in open-source software. **Software, IEEE**, v. 24, n. 1, p. 58-64, 2007. ISSN 0740-7459.

ACKROYD, S. Critical realism, organization theory, methodology, and the emerging science of reconfiguration. **Elements of a Philosophy of Management and Organization**, p. 47-77, 2010.

ÅGERFALK, P. J.; FITZGERALD, B. Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. **MIS quarterly**, p. 385-409, 2008. ISSN 0276-7783.

AIS. Senior Scholars' Basket of Journals. AIS website, 2011. Disponível em: < <https://aisnet.org/general/custom.asp?page=SeniorScholarBasket> >. Acesso em: 2015, Nov, 10th.

BHASKAR, R. **A Realist Theory of Science**. England: Harvester Press, 1975.

BHASKAR, R. **A realist theory of science**. Routledge, 2013. ISBN 1134050860.

BOGERS, M. et al. The open innovation research landscape: Established perspectives and emerging themes across different levels of analysis. **Industry and Innovation**, v. 24, n. 1, p. 8-40, 2017. ISSN 1366-2716.

CARVER, B. W. Share and share alike: Understanding and enforcing open source and free software licenses. **Berkeley Technology Law Journal**, v. 20, n. 443, 2005.

CHAU, P. Y. K.; TAM, K. Y. *Factors Affecting the Adoption of Open Systems: An Exploratory Study*. **MIS Quarterly**, v. 21, n. 1, p. 1-24, 1997. ISSN 02767783. Disponível em: < [https://www.jstor.org/stable/249740?seq=1#page\\_scan\\_tab\\_contents](https://www.jstor.org/stable/249740?seq=1#page_scan_tab_contents) >.

CHENG, H. K.; LIU, Y.; TANG, Q. The impact of network externalities on the competition between open source and proprietary software. **Journal of Management Information Systems**, v. 27, n. 4, p. 201-230, 2011. ISSN 0742-1222.

CHESBROUGH, H. **Open innovation**: Boston: Harvard Business School Press 2003.



COLLIER, A. **Critical realism: an introduction to Roy Bhaskar's philosophy**. London: Verso, 1994.

COMINO, S.; MANENTI, F. M.; PARISI, M. L. From planning to mature: On the success of open source projects. **Research Policy**, v. 36, n. 10, p. 1575-1586, 2007. ISSN 0048-7333.

DI TULLIO, D.; STAPLES, D. S. The governance and control of open source software projects. **Journal of Management Information Systems**, v. 30, n. 3, p. 49-80, 2013. ISSN 0742-1222.

DOBSON, P.; MYLES, J.; JACKSON, P. Making the case for critical realism: Examining the implementation of automated performance management systems. **Information Resources Management Journal**, v. 20, n. 2, p. 138, 2007. ISSN 1040-1628.

EASTON, G. Critical realism in case study research. **Industrial marketing management**, v. 39, n. 1, p. 118-128, 2010. ISSN 0019-8501.

ELO, S.; KYNGÄS, H. The qualitative content analysis process. **Journal of advanced nursing**, v. 62, n. 1, p. 107-115, 2008. ISSN 1365-2648.

FANG, Y.; NEUFELD, D. Understanding sustained participation in open source software projects. **Journal of Management Information Systems**, v. 25, n. 4, p. 9-50, 2009. ISSN 0742-1222.

GAMBARDELLA, A.; HALL, B. H. Proprietary versus public domain licensing of software and research products. **Research Policy**, v. 35, n. 6, p. 875-892, 2006. ISSN 0048-7333.

GASSMANN, O.; ENKEL, E.; CHESBROUGH, H. The future of open innovation. **R&d Management**, v. 40, n. 3, p. 213-221, 2010. ISSN 1467-9310.

GREWAL, R.; LILIEN, G. L.; MALLAPRAGADA, G. Location, location, location: How network embeddedness affects project success in open source systems. **Management Science**, v. 52, n. 7, p. 1043-1056, 2006. ISSN 0025-1909.

GWEBU, K. L.; WANG, J. *Adoption of Open Source Software: The role of social identification*. **Decision Support Systems**, v. 51, n. 1, p. 220-229, 2011. ISSN 0167-9236.

HERBSLEB, J. D.; MOCKUS, A. An empirical study of speed and communication in globally distributed software development. **Software Engineering, IEEE Transactions on**, v. 29, n. 6, p. 481-494, 2003. ISSN 0098-5589.

HOWISON, J.; CROWSTON, K. Collaboration Through Open Superposition: A Theory of the Open Source Way. **Mis Quarterly**, v. 38, n. 1, p. 29-50, 2014. ISSN 0276-7783.

HSIEH, H.-F.; SHANNON, S. E. Three approaches to qualitative content analysis. **Qualitative health research**, v. 15, n. 9, p. 1277-1288, 2005. ISSN 1049-7323.

KAUFFMAN, R. J.; MOHTADI, H. *Proprietary and open systems adoption in e-procurement: a risk-augmented transaction cost perspective*. Journal of Management Information Systems, v. 21, n. 1, p. 137-166, 2004. ISSN 0742-1222.

KE, W.; ZHANG, P. The effects of extrinsic motivations and satisfaction in open source software development. **Journal of the Association for Information Systems**, v. 11, n. 12, p. 784-808, 2010. ISSN 1536-9323.

KHOO, H. M.; ROBEY, D. Deciding to upgrade packaged software: a comparative case study of motives, contingencies and dependencies. **European Journal of Information Systems**, v. 16, n. 5, p. 555-567, 2007. ISSN 0960-085X.

KUEHNEL, A.-K. Microsoft, Open Source and the software ecosystem: of predators and prey—the leopard can change its spots. **Information & Communications Technology Law**, v. 17, n. 2, p. 107-124, 2008. ISSN 1360-0834.

LERNER, J.; TIROLE, J. The scope of open source licensing. **Journal of Law, Economics, and Organization**, v. 21, n. 1, p. 20-56, 2005. ISSN 8756-6222.

LIN, L. Impact of user skills and network effects on the competition between open source and proprietary software. **Electronic Commerce Research and Applications**, v. 7, n. 1, p. 68-81, 2008. ISSN 1567-4223.

LONG, J. Understanding the role of core developers in open source software development. **Journal of Information, Information Technology, and Organizations**, v. 1, p. 75-85, 2006. ISSN 1557-1319.

LÓPEZ, J.; POTTER, G. **After postmodernism: An introduction to critical realism**. A&C Black, 2005. ISBN 1847141064.

MARKUS, M. L.; SILVER, M. S. A foundation for the study of IT effects: A new look at DeSanctis and Poole's concepts of structural features and spirit. **Journal of the Association for Information systems**, v. 9, n. 10/11, p. 609, 2008. ISSN 1536-9323.

MIDHA, V.; BHATTACHERJEE, A. Governance practices and software maintenance: A study of open source projects. **Decision Support Systems**, v. 54, n. 1, p. 23-32, 2012. ISSN 0167-9236.

MILES, M. B.; HUBERMAN, A. M. **Qualitative data analysis: An expanded sourcebook**. sage, 1994. ISBN 0803955405.

MINGERS, J. Combining IS research methods: towards a pluralist methodology. **Information systems research**, v. 12, n. 3, p. 240-259, 2001. ISSN 1047-7047.

MORTON, P. Using critical realism to explain strategic information systems planning. **JITTA: Journal of Information Technology Theory and Application**, v. 8, n. 1, p. 1, 2006. ISSN 1532-4516.

MUTCH, A. Critical realism, managers and information. **British Journal of Management**, v. 10, n. 4, p. 323-333, 1999. ISSN 1467-8551.

O'MAHONY, S.; FERRARO, F. The emergence of governance in an open source community. **Academy of Management Journal**, v. 50, n. 5, p. 1079-1106, 2007. ISSN 0001-4273.

PEARCE, F.; FRAULEY, J. **Critical realism and the social sciences: Heterodox elaborations**. Univ of Toronto Pr, 2007. ISBN 0802092152.

REED, M. Critical realism: Philosophy, method, or philosophy in search of a method. **The SAGE handbook of organizational research methods**, p. 430-448, 2009.

ROUSINOPOULOS, A.-I.; ROBLES, G.; GONZÁLEZ-BARAHONA, J. M. Sentiment analysis of free/open source developers: preliminary findings from a case study. **Revista Eletrônica de Sistemas de Informação**, v. 13, n. 2, p. 1, 2014. ISSN 1677-3071.

SANTOS, C. et al. The attraction of contributors in free and open source software projects. **The Journal of Strategic Information Systems**, v. 22, n. 1, p. 26-45, 2013. ISSN 0963-8687.

SAYER, A. **Method in social science: revised 2nd edition**. Routledge, 2010. ISBN 1136961917.

SCHROLL, A.; MILD, A. A critical review of empirical research on open innovation adoption. **Journal für Betriebswirtschaft**, v. 62, n. 2, p. 85-118, 2012. ISSN 0344-9327.

SEN, R. A strategic analysis of competition between open source and proprietary software. **Journal of Management Information Systems**, v. 24, n. 1, p. 233-257, 2007. ISSN 0742-1222.

SHAFTO, P. Why big tech companies are open-sourcing their AI systems. The Conversation Website, 2016.

STAMELOS, I. et al. Code quality analysis in open source software development. **Information Systems Journal**, v. 12, n. 1, p. 43-60, 2002. ISSN 1365-2575.

STEINMACHER, I. F. et al. HISTORICAL ANALYSIS OF MESSAGE CONTENTS TO RECOMMEND ISSUES TO OPEN SOURCE SOFTWARE CONTRIBUTORS. **Revista Eletrônica de Sistemas de Informação**, v. 13, n. 2, 2014. ISSN 1677-3071.

STEWART, K. J.; AMMETER, A. P.; MARUPING, L. M. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. **Information Systems Research**, v. 17, n. 2, p. 126-144, 2006. ISSN 1047-7047.

STEWART, K. J.; GOSAIN, S. The impact of ideology on effectiveness in open source software development teams. **Mis Quarterly**, p. 291-314, 2006. ISSN 0276-7783.

STRAUSS, A.; CORBIN, J. **Basics of qualitative research techniques**. Sage publications, 1998.

SUBRAMANIAM, C.; SEN, R.; NELSON, M. L. Determinants of open source software project success: A longitudinal study. **Decision Support Systems**, v. 46, n. 2, p. 576-585, 2009. ISSN 0167-9236.

TSAI, J. For Better or Worse: Introducing the GNU General Public License Version 3. **Berkeley Technology Law Journal**, v. 23, n. 1, p. 547-581, 2008. ISSN 1086-3818.

VON KROGH, G. et al. Carrots and rainbows: Motivation and social practice in open source software development. **MIS Quarterly**, v. 36, n. 2, p. 649-676, 2012. ISSN 0276-7783.

WEST, J.; GALLAGHER, S. Challenges of open innovation: the paradox of firm investment in open-source software. **R&d Management**, v. 36, n. 3, p. 319-331, 2006. ISSN 1467-9310.

WYNN JR, D.; WILLIAMS, C. K. Principles for Conducting Critical Realist Case Study Research in Information Systems. **MIS Quarterly**, v. 36, n. 3, 2012. ISSN 0276-7783.

YIN, R. K. **Case study research: design and methods**. Sage publications, 2014. ISBN 1483302008.

ZACHARIADIS, M.; SCOTT, S. V.; BARRETT, M. I. Designing mixed-method research inspired by a critical realism philosophy: a tale from the field of IS innovation. 2012, Association for Information Systems.